*Załącznik nr 1 do Uchwały nr \_\_\_\_\_\_\_ - 2018/2019 z dnia 25 marca 2019 r.*

*w sprawie wytycznych dla tworzenia i zmian programów studiów pierwszego stopnia, drugiego stopnia*

*oraz jednolitych studiów magisterskich rozpoczynających się od roku akademickiego 2019/2020.*

Opis **zajęć (sylabus)**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nazwa zajęć: | | Programowanie komponentowe | | | | | | | **ECTS** | **4** |
| Nazwa zajęć w j. angielskim: | | Component programming | | | | | | | | |
| Zajęcia dla kierunku studiów: | | **Informatyka** | | | | | | | | |
|  | |  | | | | | | | | |
| Język wykładowy: | | polski | | | | Poziom studiów: | | studia I stopnia | | |
| Forma studiów: | stacjonarne  niestacjonarne | Status zajęć: |  podstawowe   kierunkowe | obowiązkowe  do wyboru | | Numer semestru: ……**5**….. | |  semestr zimowy   semestr letni | | |
|  |  | Rok akademicki, od którego obowiązuje opis (rocznik): | | | | 2021/2022 | Numer katalogowy: | **ZIM-IN-1Z-05Z-32\_2** | | |
|  | | | | | | | | | | |
| Koordynator zajęć: | |  | | | | | | | | |
| Prowadzący zajęcia: | |  | | | | | | | | |
| Jednostka realizująca: | |  | | | | | | | | |
| Jednostka zlecająca: | |  | | | | | | | | |
| Założenia, cele i opis zajęć: | | Celem przedmiotu jest zapoznanie studentów z podstawami tworzenia programów w technologiach komponentowych. W trakcie zajęć studenci poznają możliwości wykorzystania zdefiniowanych przez siebie komponentów w programach napisanych w wybranym języku programowania.Tematyka zajęć:1. Przypomnienie podstaw programowania proceduralnego w wybranym języku programowania: wprowadzanie i wyprowadzanie danych, instrukcja warunkowa, instrukcja iteracyjna WHILE, instrukcje iteracyjne FOR, sekwencje – łańcuchy znaków, krotki, listy i słowniki, funkcje i ich wykorzystanie do tworzenia własnych modułów i pakietów, programy z argumentami pozycyjnymi, obsługa wyjątków, operacje na plikach.2. Przypomnienie podstaw programowania obiektowego w wybranym języku programowania: klasy, metody, obiekty i pola, konstruktory i atrybuty klas, polimorfizm i dziedziczenie.3. Tworzenie interfejsów graficznych GUI z wykorzystaniem wybranego języka programowania jako przykład wykorzystania paradygmatu programowania obiektowego. Wykorzystanie elementów graficznych i multimedialnych.4. Pojęcie interfejsu. Przykłady wykorzystania interfejsów w programowaniu obiektowym.5. Wprowadzenie do programowania komponentowego. Teoria i inżynieria oprogramowania komponentowego. Wzorzec odwróconego sterowania. Pojęcie i rola kontenera,6. Przegląd wybranych technologii komponentowych.7. Wykorzystanie odpowiedniego pakietu (biblioteki) wybranego języka programowania do tworzenia komponentów. Typy komponentów i ich użycie. Cykl życia komponentów. Wersjonowanie komponentów.8. Tworzenie aplikacji w wybranej technologii komponentowej. | | | | | | | | |
| Formy dydaktyczne, liczba godzin: | | a) wykład ; liczba godziin ...9…  b) ćwiczenia laboratoryjne; liczba godzin ...18... | | | | | | | | |
| Metody dydaktyczne: | | wykład, dyskusja problemu, rozwiązywanie problemu, studium przypadku, projekt grupowy, konsultacje | | | | | | | | |
| Wymagania formalne  i założenia wstępne: | | Wymagana jest wiedza z zakresu przedmiotów: Wstęp do programowania, Programowanie imperatywno – strukturalne, Programowanie obiektowe, Algorytmy i struktury danych, Inżynieria oprogramowania | | | | | | | | |
| Efekty uczenia się: | | Wiedza:  1 – zna specyfiki programowania proceduralnego, obiektowego i komponentowego w wybranym języku programowania  2 – rozumie pojęcie komponentu programowego oraz ma wiedzę na temat podstaw tworzenia i wykorzystania wybranej technologii komponentowej | | | Umiejętności:  1 – potrafi tworzyć programy z wykorzystaniem paradygmatów: proceduralnego, obiektowego i komponentowego, w tym projektować, wytwarzać, wdrażać i utrzymywać komponenty programowe, wykorzystując wybraną technologię komponentową  2 - projektuje rozbudowane aplikacje z wykorzystaniem stworzonych samodzielnie obiektów i komponentów, w tym modułów wielokrotnego użycia, wykorzystując wybrany język programowania  3 – umie znajdować i niwelować zależności pomiędzy częściami oprogramowania | | | | Kompetencje: | |
| Sposób weryfikacji efektów uczenia się: | | Pisemny test wiedzy ze znajomości specyfiki programowania proceduralnego, obiektowego oraz komponentowego w wybranym języku programowania (efekty wiedzy)  Kolokwium praktyczne polegające na napisaniu określonej liczby programów komputerowych rozwiązujących postawione problemy (efekt umiejętności 1)  Grupowy projekt aplikacji wykorzystującej interfejs graficzny (GUI) oraz technologię komponentową (efekty umiejętności 2 oraz 3) | | | | | | | | |
| Forma dokumentacji osiągniętych efektów uczenia się: | | Ocenione prace studentów z pisemnego testu wiedzy  Ocenione pliki kolokwium praktycznego w wersji elektronicznej  Ocenione i zarchiwizowane pliki grupowych projektów studenckich | | | | | | | | |
| Elementy i wagi mające wpływ  na ocenę końcową: | | **Kolokwium praktyczne – 50%**  **Projekt grupowy – 25%**  **Test wiedzy – 25%** | | | | | | | | |
| Miejsce realizacji zajęć: | | laboratorium komputerowe | | | | | | | | |
| Literatura podstawowa i uzupełniająca:  Literatura podstawowa:  - Szyperski Clemens: Oprogramowanie komponentowe. Obiekty to za mało, WNT 2001.  - Andy Ju An Wang, Kai Qian: Component-Oriented Programming, John Wiley & Sons, 2005.  - Ian Sommerville: Inżynieria oprogramowania, WNT 2003.  Literatura uzupełniająca:  - Michael Dawson: Python dla każdego. Podstawy programowania. Od zera do bohatera, Helion 2014.  - Dive into Python – internetowy podręcznik programowania w Pythonie  - materiały prezentacyjne w postaci plików udostępniane przez prowadzącego | | | | | | | | | | |
| UWAGI  Minimalna liczba punktów konieczna do zaliczenia: 50% z kolokwium praktycznego, 50% z projektu indywidualnego oraz 50% z testu wiedzy | | | | | | | | | | |

Wskaźniki ilościowe charakteryzujące moduł/przedmiot:

|  |  |
| --- | --- |
| Szacunkowa sumaryczna liczba godzin pracy studenta (kontaktowych i pracy własnej) niezbędna dla osiągnięcia zakładanych dla zajęć efektów uczenia się - na tej podstawie należy wypełnić pole ECTS: | **120 h** |
| Łączna liczba punktów ECTS, którą student uzyskuje na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich lub innych osób prowadzących zajęcia: | **0.9 ECTS** |

Tabela zgodności kierunkowych efektów uczenia się z efektami przedmiotu:

|  |  |  |  |
| --- | --- | --- | --- |
| kategoria efektu | Efekty uczenia się dla zajęć: | Odniesienie do efektów dla programu studiów dla kierunku | Oddziaływanie zajęć na efekt kierunkowy\*) |
| Wiedza 1 | zna specyfiki programowania proceduralnego, obiektowego i komponentowego w wybranym języku programowania | K\_W08/ P6S\_WG | 2 |
| Wiedza 2 | rozumie pojęcie komponentu programowego oraz ma wiedzę na temat podstaw tworzenia i wykorzystania wybranej technologii komponentowej | K\_W08/ P6S\_WG | 2 |
| Umiejętności 1 | potrafi tworzyć programy z wykorzystaniem paradygmatów: proceduralnego, obiektowego i komponentowego, w tym projektować, wytwarzać, wdrażać i utrzymywać komponenty programowe, wykorzystując wybraną technologię komponentową | K\_U10/ P6S\_UW | 1 |
| Umiejętności 2 | projektuje rozbudowane aplikacje z wykorzystaniem stworzonych samodzielnie obiektów i komponentów, w tym modułów wielokrotnego użycia, wykorzystując wybrany język programowania | K\_U04/ P6S\_UK  K\_U17/ P6S\_UW  K\_U18/ P6S\_UW  K\_U20/ P6S\_UW | 2  1  1  2 |
| Umiejętności 3 | umie znajdować i niwelować zależności pomiędzy częściami oprogramowania | K\_U20/ P6S\_UW | 2 |

\*)

3 – zaawansowany i szczegółowy,

2 – znaczący,

1 – podstawowy,